PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 17/27 | A1

(11) International Publication Number: WO 99/62001

(43) International Publication Date: 2 December 1999 (02.12.99)

(21) International Application Number: PCT/US99/11856

(22) International Filing Date: 28 May 1999 (28.05.99)

(30) Priority Data:
09/087,468 29 May 1998 (29.05.98) US

(81) Designated States: CA, CN, JP, KR, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published

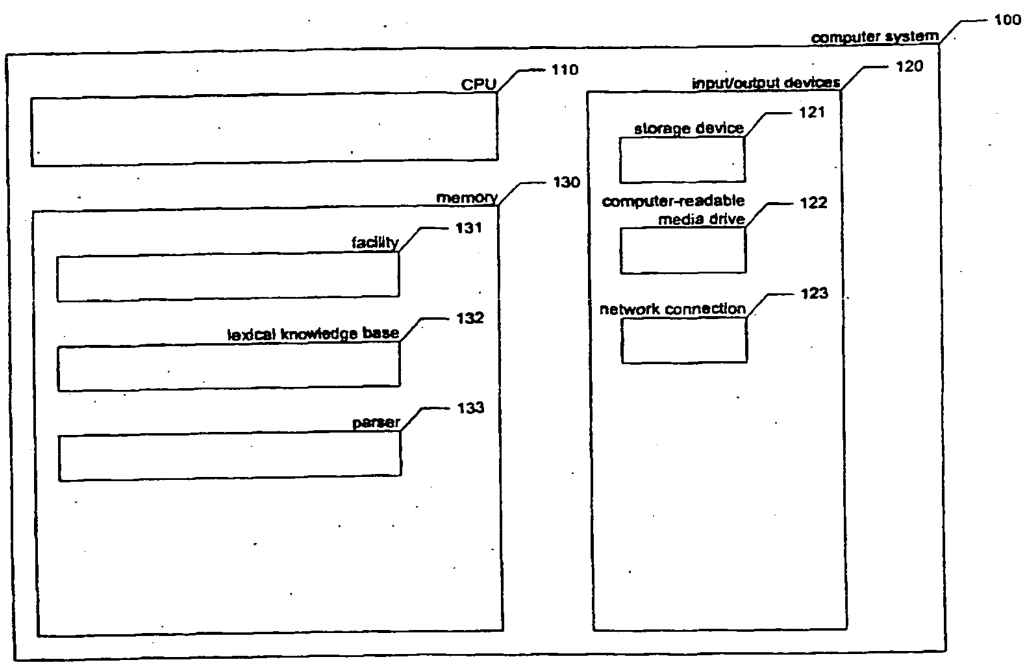
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(71) Applicant: MICROSOFT CORPORATION [US/US]; One Microsoft Way, Redmond, WA 98052-6399 (US).

(72) Inventors: WU, Andi; 2036 152nd Avenue S.E., Bellevue, WA 98007 (US). RICHARDSON, Stephen, D.; 18028 N.E. 132nd Street, Redmond, WA 98052 (US). JIANG, Zixin; 9307 176th Place N.E. #3, Redmond, WA 98052 (US).

(74) Agents: KOEHLER, Steven, M. et al.; Westman, Champlin & Kelly, P.A., International Centre, Suite 1600, 900 Second Avenue South, Minneapolis, MN 55402-3319 (US).

(54) Title: WORD SEGMENTATION IN CHINESE TEXT



(57) Abstract

The present invention provides a facility for selecting from a sequence of natural language characters combinations of characters that may be words. The facility uses probability indications for each of a plurality of words as a function of adjacent characters.

NSDOCID: <WO___9962001A1_I_>

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

WORD SEGMENTATION IN CHINESE TEXT

TECHNICAL FIELD

The invention relates generally to the field of natural language processing, and, more specifically, to the field of word segmentation.

BACKGROUND OF THE INVENTION

Word segmentation refers to the process of identifying the individual words that make up an expression of language, such as text. Word segmentation is useful for checking spelling and grammar, synthesizing speech from text, and performing natural language parsing and understanding, all of which benefit from an identification of individual words.

Performing word segmentation of English text is rather straightforward, since spaces and punctuation marks generally delimit the individual words in the text. Consider the English sentence in Table 1 below.

The motion was then tabled--that is, removed
indefinitely from consideration.

Table 1

By identifying each contiguous sequence of spaces and/or punctuation marks as the end of the word preceding the sequence, the English sentence in Table 1 may be straightforwardly segmented as shown in Table 2 below.

The motion was then tabled -- that is, removed.
indefinitely from consideration.

Table 2

In Chinese text, word boundaries are implicit rather than explicit. Consider the sentence in Table 3 below, meaning "The committee discussed this problem yesterday afternoon in Buenos Aires."

INSDOCID: <WO___9962001A1_I_>

昨天下午委员会在布宜诺斯艾利斯讨论了这个问题。

Table 3

Despite the absence of punctuation and spaces from the sentence, a reader of Chinese would recognize the sentence in Table 3 as being comprised of the words separately underlined in Table 4 below.

昨天下午委员会在布宜诺斯艾利斯讨论了这个问题。

Table 4

It can be seen from the examples above that Chinese word segmentation cannot be performed in the same manner as English word segmentation. An accurate and efficient approach to automatically performing Chinese segmentation would nonetheless have significant utility.

SUMMARY OF THE INVENTION

The present invention provides a facility for selecting from a sequence of natural language characters combinations of characters that may be words. The facility uses probability indications for each of a plurality of words as a function of adjacent characters.

One aspect of the present invention is a method in a computer system for identifying individual words occurring in a sentence of text. The method includes the steps of: for each of a plurality of words, storing an indication of probability of whether the word occurs in natural language text as a function of adjacent characters; and for each of a plurality of contiguous groups of characters occurring in the sentence, determining overlapping possible words, ascertaining probability based on the stored indication and adjacent characters and submitting the groups of characters determined to be possible words to a parser with an indication of probability. A computer readable medium for storing the instructions implementing the same is also provided.

BNSDOCID: <WO___9962001A1_I_>

The second aspect of the present invention includes computer memory containing a word segmentation data structure for use in identifying individual words occurring in natural language test. The data structure includes for each of a plurality of words an indication of probability of whether the word occurs in natural language text as a function of adjacent characters.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a high-level block diagram of the general-purpose computer system upon which the facility preferably executes.

Figure 2 is an overview flow diagram showing the two phases in which the facility preferably operates.

Figure 3 is a flow diagram showing the steps preferably performed by the facility in order to augment the lexical knowledge base in the initialization phase to include information used to perform word segmentation.

Figure 4 is a flow diagram showing the steps preferably performed in order to determine whether a particular word can contain other, smaller words.

Figure 5 is a flow diagram of the steps preferably performed by the facility in order to segment a sentence into its constituent words.

Figure 6 is a flow diagram showing the steps preferably performed by the facility in order to add multiple-character words to the word list.

Figure 7 is a flow diagram showing the step preferably performed by the facility in order to test the NextChar and CharPos conditions for a word candidate.

Figure 8 is a flow diagram showing the steps preferably performed by the facility in order to determine whether the last character of the current word candidate overlaps with another word candidate that may be a word.

Figure 9 is a flow diagram showing the steps preferably performed by the facility in order to add single-character words to the word list.

Figure 10 is a flow diagram showing the steps preferably performed by the facility in order to assign probabilities to the lexical records generated from the words in the word list in accordance with a first approach.

Figure 11 is a flow diagram showing the steps preferably performed by the facility in order to assign probabilities to the lexical records generated from the words in the word list in accordance with a second approach.

Figure 12 is a parse tree diagram showing a parse tree generated by the parser representing the syntactic structure of the sample sentence.

DETAILED DESCRIPTION OF THE INVENTION

The present invention provides word segmentation in Chinese text. In a preferred embodiment, a word segmentation software facility ("the facility") provides word segmentation for text in unsegmented languages such as Chinese by (1) evaluating the possible combinations of characters in an input sentence and discarding those unlikely to represent words in the input sentence, (2) looking up the remaining combinations of characters in a dictionary to determine whether they may constitute words, and (3) submitting the combinations of characters determined to be words to a natural language parser as alternative lexical records representing the input sentence. The parser generates a syntactic parse tree representing the syntactic structure of the input sentence, which contains only those lexical records representing the combinations of characters certified to be words in the input sentence. When submitting the lexical records to the parser, the facility weights the lexical records so that longer combinations of characters, which more commonly represent the correct segmentation of a sentence than shorter combinations of characters, are considered by the parser before shorter combinations of characters.

In order to facilitate discarding combinations of characters unlikely to represent words in the input sentence, the facility adds to the dictionary, for each character occurring in the dictionary, (1) indications of all of the different combinations of word length and character position in which the word appears, and (2) indications of

all of the characters that may follow this character when this character begins a word. The facility further adds (3) indications to multiple-character words of whether sub-words within the multiple-character words are viable and should be considered. In processing a sentence, the facility discards (1) combinations of characters in which any character is used in a word length/position combination not occurring in the dictionary, and (2) combinations of characters in which the second character is not listed as a possible second character of the first character. The facility further discards (3) combinations of characters occurring in a word for which sub-words are not to be considered.

In this manner, the facility both minimizes the number of character combinations looked up in the dictionary and utilizes the syntactic context of the sentence to differentiate between alternative segmentation results that are each comprised of valid words.

Figure 1 is a high-level block diagram of the general-purpose computer system upon which the facility preferably executes. The computer system 100 contains a central processing unit (CPU) 110, input/output devices 120, and a computer memory (memory) 130. Among the input/output devices is a storage device 121, such as a hard disk drive; a computer-readable media drive 122, which can be used to install software products, including the facility, which are provided on a computer-readable medium, such as a CD-ROM; and a network connection 123, through which the computer system 100 may communicate with other connected computer systems (not shown). The memory 130 preferably contains a word segmentation facility 131 for identifying individual words occurring in Chinese text, a syntactic parser 133 for generating a parse tree representing the syntactic structure of a sentence of natural language text from lexical records representing the words occurring in the natural language text, and a lexical knowledge base 132 for use by the parser in constructing lexical records for a parse tree and for use by the facility to identify words occurring in natural language text. While the facility is preferably implemented on a computer system configured as described above, those

skilled in the art will recognize that it may also be implemented on computer systems having different configurations.

Figure 2 is an overview flow diagram showing the two phases in which the facility preferably operates. In step 201, as part of an initialization phase, the facility augments a lexical knowledge base to include information used by the facility to perform word segmentation. Step 201 is discussed in greater detail below in conjunction with Figure 3. Briefly, in step 201, the facility adds entries to the lexical knowledge base for the characters occurring in any word in the lexical knowledge base. The entry added for each character includes a CharPos attribute that indicates the different positions at which the character appears in words. The entry for each character further contains a NextChars attribute that indicates the set of characters that occur in the second position of words that begin with the current character. Finally, the facility also adds an IgnoreParts attribute to each word occurring in the lexical knowledge base that indicates whether the sequence of characters comprising the word should ever be considered to comprise smaller words that together make up the current word.

After step 201, the facility continues in step 202, ending the initialization phase and beginning the word segmentation phase. In the word segmentation phase, the facility uses the information added to the lexical knowledge base to perform word segmentation of sentences of Chinese text. In step 202, the facility receives a sentence of Chinese text for word segmentation. In step 203, the facility segments the received sentence into its constituent words. Step 203 is discussed in greater detail below in conjunction with Figure 5. Briefly, the facility looks up in the lexical knowledge base a small fraction of all the possible contiguous combinations of characters in the sentence. The facility then submits to a syntactic parser the looked-up combinations of characters that are indicated to be words by the lexical knowledge base. The parser, in determining the syntactic structure of the sentence, identifies the combinations of characters intended to comprise words in the sentence by its author. After step 203, the facility continues at step 202 to receive the next sentence for word segmentation.

Figure 3 is a flow diagram showing the steps preferably performed by the facility in order to augment the lexical knowledge base in the initialization phase to include information used to perform word segmentation. These steps (a) add entries to the lexical knowledge base for the characters occurring in words in the lexical knowledge base; (b) add CharPos and NextChars attributes to the character entries in the lexical knowledge base; (c) add the IgnoreParts attribute to the entries for words in the lexical knowledge base.

In steps 301-312, the facility loops through each word entry in the lexical knowledge base. In step 302, the facility loops through each character position in the word. That is, for a word containing three characters, the facility loops through the first, second, and third characters of the word. In step 303, if the character in the current character position has an entry in the lexical knowledge base, then the facility continues in step 305, else the facility continues in step 304. In step 304, the facility adds an entry to the lexical knowledge base for the current character. After step 304, the facility continues in step 305. In step 305, the facility adds an ordered pair to the CharPos attribute stored in the character's entry in the lexical knowledge base to indicate that the character may occur in the position in which it occurs in the current word. The ordered pair added has the form (*position*, *length*). where *position* is the position that the character occupies in the word and *length* is the number of characters in the word. For example, for the character "委" in the word "委 员 会," the facility will add the ordered pair (1, 3) to the list of ordered pairs stored in the CharPos attribute in the lexical knowledge base entry for the character "委." The facility preferably does not add the ordered pair as described in step 305 if the ordered pair is already contained in the CharPos attribute for the current word. In step 306, if additional characters remain in the current word to be processed, then the facility continues in step 302 to process the next character, else the facility continues in step 307.

In step 307, if the word is a single character word, then the facility continues in step 309, else the facility continues in step 308. In step 308, the facility adds

a character in the second position of the current word to the list of characters in the NextChars attribute in the lexical knowledge base record for the character in the first position of the current word. For example, for the word "委员会," the facility adds the character "员" to the list of characters stored for the NextChars attribute of the character "委." After step 308, the facility continues in step 309.

In step 309, if the current word can contain other, smaller words, then the facility continues in step 311, else the facility continues in step 310. Step 309 is discussed in further detail below in conjunction with Figure 4. Briefly, the facility employs a number of heuristics to determine whether an occurrence of the sequence of characters that make up the current word may in some context make up two or more smaller words.

In step 310, the facility sets an IgnoreParts attribute for the word in the lexical knowledge base entry for the word. Setting the IgnoreParts attribute indicates that, when the facility encounters this word in a sentence of input text, it should not perform further steps to determine whether this word contains smaller words. After step 310, the facility continues in step 312. In step 311, because the current word can contain other words, the facility clears the IgnoreParts attribute for the word, so that the facility, when it encounters the word in a sentence of input text, proceeds to investigate whether the word contains smaller words. After step 311, the facility continues in step 312. In step 312, if additional words remain in the lexical knowledge base to be processed, then the facility continues in step 301 to process the next word, else these steps conclude.

When the facility performs the steps shown in Figure 3 to augment the lexical knowledge base by assigning CharPos and NextChars attributes to each character, it assigns these attributes to the characters occurring in the sample sentence shown in Table 3 as shown below in Table 5.

Character	CharPos	NextChars
昨	(1,2) (1,3) (3,4)	儿 天 晚

Character	CharPos	NextChars
天	(1,2) (2,2) (1,3) (2,3) (3,3) (1,4) (2,4) (3,4) (4,4)	安 崩 兵 …
下	(1,2) (2,2) (1,3) (2,3) (3,3) (1,4) (2,4) (3,4) (4,4)	巴 百 班 … 午 …
午	(1,2) (2,2) (2,3) (2,4)	餐 饭 后 …
委	(1,2) (2,2) (1,3) (2,3) (3,3) (1,4) (3,4) (4,4) (3,5)	靡 派 屈 … 员 …
员	(1,2) (2,2) (2,3) (3,3) (2,4) (3,4) (4,4)	额 工 司 外
会	(1,2) (2,2) (1,3) (2,3) (3,3) (1,4) (2,4) (3,4) (4,4) (4,5)	标 餐 操 …
在	(1,2) (2,2) (1,3) (2,3) (1,4) (2,4) (3,4) (4,4)	案 场 朝 …
布	(1,2) (2,2) (1,3) (2,3) (3,3) (1,4) (2,4) (3,4) (4,4) (1,5) (2,5) (3,5) (4,5) (1,6) (2,6) (1,7)	达 店 丁 … 宜 …
宜	(1,2) (2,2) (2,3) (3,3) (2,4) (3,4) (4,4) (3,6) (2,7)	宾 昌 城 …
诺	(1,2) (2,2) (1,3) (2,3) (3,3) (2,4) (3,4) (4,4) (3,7)	贝 丁 曼 萨 言
斯	(1,2) (2,2) (1,3) (2,3) (3,3) (1,4) (2,4) (3,4) (4,4) (1,5) (2,5) (3,5) (4,5) (5,5) (1,6) (3,6) (4,6) (5,6) (6,6) (4,7) (5,7) (6,7) (7;7)	文 德 拉 …
艾	(1,2) (2,2) (1,3) (3;4) (4,4) (1,5) (5,7)	比 丁 黎 …
利	(1,2) (2,2) (1,3) (2,3) (3,3) (1,4) (2,4) (3,4) (4,4) (2,5) (3,5) (4,5) (5,6) (6,7)	薄 比 弊 …
斯	(1,2) (2,2) (1,3) (2,3) (3,3) (1,4) (2,4) (3,4) (4,4) (1,5) (2,5) (3,5) (4,5) (5,5) (1,6) (3,6) (4,6) (5,6) (6,6) (4,7) (5,7) (6,7) (7,7)	文 德 拉 …
讨	(1,2) (2,2) (1,3) (2,3) (1,4) (2,4)	伐 饭 好 价 叫 论 人 厌
论	(1,2) (2,2) (1,3) (2,3) (3,3) (1,4) (2,4) (3,4) (4,4)	处 点 调 …
了	(1,2) (2,2) (1,3) (3,3) (1,4) (2,4) (3,4) (4,4)	不 结 解 …

Character	CharPos	NextChars
这	(1,2) (1,3) (1,4)	边 儿 个 …
个	(1,2) (2,2) (1,3) (2,3) (3,3) (1,4) (2,4)	别 儿 旧 …
问	(1,2) (2,2) (1,3) (2,3) (1,4) (3,4) (4,4)	长 答 道 … 题 …
题	(1,2) (2,2) (2,3) (3,3) (2,4) (4,4)	跋 材 词 …

Table 5: Character Lexical Knowledge Base Entries

It can be seen from Table 5, for instance, from the CharPos attribute of the character "昨" that this character can appear as the first character of words that are 2, 3, or 4 characters long. It can further be seen from the NextChars attribute of the character "昨" that, in words beginning with this character, the second character may be either "儿," "天," or "晚."

Figure 4 is a flow diagram showing the steps preferably performed in order to determine whether a particular word can contain other, smaller words. As an analogy to English, if spaces and punctuation characters were removed from an English sentence, the sequence of characters "beat" could be interpreted either as the word "beat" or as the two words "be" and "at." In step 401, if the word contains four or more characters, then the facility continues in step 402 to return the result that the word cannot contain other words, else the facility continues in step 403. In step 403, if all the characters in the word can constitute single-character words, then the facility continues in step 405, else the facility continues in step 404 to return the result that the word cannot contain other words. In step 405, if the word contains a word frequently used as a derivational affix, that is, a prefix or a suffix, then the facility continues in step 406 to return the result that the word cannot contain other words, else the facility continues in step 407. In step 407, if an adjacent pair of characters in the word are often divided when they appear adjacently in text of the language, then the facility continues in step 409 to return the result that the word can contain other words, else the facility continues in step 408 to return the result that the word cannot contain other words.

The results of determining whether particular words can contain other, smaller words are shown below in Table 6.

Word	IgnoreParts
昨天	set
天下	clear
下午	set
委员会	clear
委员	set
布宜诺斯艾利斯	set
讨论	set
这个	clear
问题	set

Table 6: Word Lexical Knowledge Base Entries

For example, it can be seen from Table 6 that the facility has determined that the word "昨天" cannot contain other words, while the word "天下" may contain other words.

Figure 5 is a flow diagram of the steps preferably performed by the facility in order to segment a sentence into its constituent words. These steps generate a word list identifying different words of the language that occur in the sentence. The word list is then submitted to a parser to identify the subset of words in the word list that were intended to comprise the sentence by its author.

In step 501, the facility adds to the word list multiple-character words occurring in the sentence. Step 501 is discussed in greater detail below in conjunction with Figure 6. In step 502, the facility adds to the word list the single-character words occurring in the sentence. Step 502 is discussed in greater detail below in conjunction with Figure 9. In step 503, the facility generates lexical records used by the lexical parser for the words that have been added to the word list in steps 501 and 502. In step 504, the

facility assigns probabilities to the lexical records. The probability of a lexical record reflects the likelihood that the lexical record will be part of a correct parse tree for the sentence, and is used by the parser to order the application of the lexical records in the parsing process. The parser applies the lexical records during the parsing process in decreasing order of their probabilities. Step 504 is discussed in greater detail below in conjunction with Figure 10. In step 505, the facility utilizes the syntactic parser to parse the lexical records in order to produce a parse tree reflecting the syntactic structure of the sentence. This parse tree has a subset of the lexical records generated in step 503 as its leaves. In step 506, the facility identifies as words of the sentence the words represented by the lexical records that are the leaves of the parse tree. After step 506, these steps conclude.

Figure 6 is a flow diagram showing the steps preferably performed by the facility in order to add multiple-character words to the word list. These steps use a current position within the sentence in analyzing the sentence to identify multiple-character words. These steps further utilize the CharPos, NextChar, and IgnoreParts attributes added to the lexical knowledge base by the facility as shown in Figure 4. In accordance with a first preferred embodiment, the facility retrieves these attributes from a lexical knowledge base on an as-needed basis during the performance of the steps shown in Figure 6. In a second preferred embodiment, the values of the NextChar attributes and/or the CharPos attributes of the characters in the sentence are all pre-loaded before the performance of the steps shown in Figure 6. In conjunction with the second preferred embodiment, a 3-dimensional array is preferably stored in the memory that contains the value of the CharPos attribute for each character occurring in the sentence. This array indicates, for a character at a given position in the sentence, whether the character may be at a given position in a word of a given length. Caching the values of these attributes allows them to be officially accessed when performing the steps shown in Figure 6.

BNSDOCID: <WO__9962001A1_I_>

In step 601, the facility sets this position at the first character of the sentence. In step 602-614, the facility continues to repeat steps 603-613 until the position has advanced to the end of the sentence.

In steps 603-609, the facility loops through each word candidate that begins at the current position. The facility preferably begins with the word candidate that starts at the current position and is seven characters long, and, in each iteration, removes one character from the end of the word candidate until the word candidate is two characters long. If there are fewer than seven characters remaining in the sentence beginning from the current position, the facility preferably omits the iterations for the word candidates for which there are insufficient characters remaining in the sentence. In step 604, the facility tests for the current word candidate conditions relating to the NextChar and CharPos attributes of the characters comprising the word candidate. Step 604 is discussed in greater detail below in conjunction with Figure 7. If both the NextChar and CharPos conditions are satisfied for the word candidate, then the facility continues in step 605, else the facility continues in step 609. In step 605, the facility looks up the word candidate in the lexical knowledge base to determine whether the word candidate is a word. In step 606, if the word candidate is a word, then the facility continues in step 607, else the facility continues in step 609. In step 607, the facility adds the word candidate to the list of words occurring in the sentence. In step 608, if the word candidate may contain other words, *i.e.*, if the IgnoreParts attribute for the word is clear, then the facility continues in step 609, else the facility continues in step 611. In step 609, if additional word candidates remain to processed, then the facility continues in step 603 to process the next word candidate, else the facility continues in step 610. In step 610, the facility advances the current position one character toward the end of the sentence. After step 610, the facility continues in step 614.

In step 611, if the last character of the word candidate overlaps with another word candidate that may also be a word, then the facility continues in step 613, else the facility continues in step 612. Step 611 is discussed in greater detail below in

conjunction with Figure 8. In step 612, the facility advances the position to the character in the sentence after the last character of the word candidate. After step 612, the facility continues in step 614. In step 613, the facility advances the position to the last character of the current word candidate. After step 613, the facility continues in step 614. In step 614, if the position is not at the end of the sentence, then the facility continues in step 602 to consider a new group of word candidates, else these steps conclude.

Figure 7 is a flow diagram showing the step preferably performed by the facility in order to test the NextChar and CharPos conditions for a word candidate. In step 701, if the second character of the word candidate is in the NextChar list of the first character of the word candidate, then the facility continues in step 703, else the facility continues in step 702 to return the result that the conditions are both satisfied. In steps 703-706 the facility loops through each character position in the word candidate. In step 704, if the ordered pair made up of the current position and the length of the word candidate is among the ordered pairs in the CharPos list for the character in the current character position, then the facility continues in step 706, else the facility continues in step 705 to return the result that the conditions are not both satisfied. In step 706, if additional character positions remain in the word candidate to be processed, then the facility continues in step 703 to process the next character position in the word candidate, else the facility continues in step 707 to return the result that both conditions are satisfied by the word candidate.

Figure 8 is a flow diagram showing the steps preferably performed by the facility in order to determine whether the last character of the current word candidate overlaps with another word candidate that may be a word. In step 801, if the character after the word candidate is in the list of characters in the NextChar attribute for the last character of the word candidate, then the facility continues in step 803, else the facility continues in step 802 to return the result that there is no overlap. In step 803, the facility looks up in the lexical knowledge base the word candidate without its last character in order to determine whether the word candidate without its last character is a word. In

step 804, if the word candidate without its last character is a word, then the facility continues in step 806 to return the result that there is overlap, else the facility continues in step 805 to return the result that there is no overlap.

The performance of the steps shown in Figure 6 with respect to the example as shown below in Table 7.

number	combination	CharPos	NextChars	look up?	is a word?
1	昨天下午委员会	fail on 昨	pass	no	no
2	昨天下午委员	fail on 昨	pass	no	no
3	昨天下午委	fail on 昨	pass	no	no
4	昨天下午	fail on 昨	pass	no	no
5	昨天下	pass	pass	yes	no
6	昨天	pass	pass	yes	yes
7	天下午委员会在	fail on 天	pass	no	no
8	天下午委员会	fail on 天	pass	no	no
9	天下午委员	fail on 天	pass	no	no
10	天下午委	fail on 午	pass	no	no
11	天下午	fail on 午	pass	no	no
12	天下	pass	pass	yes	yes
13	下午委员会在布	fail on 下	pass	no	no
14	下午委员会在	fail on 下	pass	no	no
15	下午委员会	fail on 下	pass	no	no
16	下午委员	pass	pass	yes	no
17	下午委	pass	pass	yes	no
18	下午	pass	pass	yes	yes
19	委员会在布宜诺	fail on 委	pass	no	no
20	委员会在布宜	fail on 委	pass	no	no
21	委员会在布	fail on 委	pass	no	no

number	combination	CharPos	NextChars	look up?	is a word?
22	委员会讨	fail on 讨	pass	no	no
23	委员会	pass	pass	yes	yes
24	委员	pass	pass	yes	yes
25	会在布宜诺斯艾	fail on 会	fail	no	no
26	会在布宜诺斯	fail on 会	fail	no	no
27	会在布宜诺	fail on 会	fail	no	no
28	会在布宜	pass	fail	no	no
29	会在布	pass	fail	no	no
30	会在	pass	fail	no	no
31	在布宜诺斯艾利	fail on 在	fail	no	no
32	在布宜诺斯艾	fail on 在	fail	no	no
33	在布宜诺斯	fail on 在	fail	no	no
34	在布宜诺	pass	fail	no	no
35	在布宜	pass	fail	no	no
36	在布	pass	fail	no	no
37	布宜诺斯艾利斯	pass	pass	yes	yes
38	讨论了这个问题	fail on 讨	pass	no	no
39	讨论了这个问	fail on 讨	pass	no	no
40	讨论了这个	fail on 讨	pass	no	no
41	讨论了这	fail on 这	pass	no	no
42	讨论了	pass	pass	yes	no
43	讨论	pass	pass	yes	yes
44	了这个问题	fail on 了	fail	no	no
45	了这个问	fail on 这	fail	no	no
46	了这个	fail on 这	fail	no	no
47	了这	fail on 这	fail	no	no
48	这个问题	pass	pass	yes	no
49	这个问	fail on 问	pass	no	no

BNSDOCID: <WO___9962001A1_I_>

number	combination	CharPos	NextChars	look up?	is a word?
50	这个	pass	pass	yes	yes
51	个问题	pass	fail	no	no
52	个问	pass	fail	no	no
53	问题	pass	pass	yes	yes

Table 7: Character Combinations Considered

Table 7 indicates, for each of the 53 combinations of characters from the sample sentence considered by the facility: the result of the CharPos test, the result of the NextChars test, whether the facility looked up the word in the lexical knowledge base, and whether the lexical knowledge base indicated that the combination of characters is a word.

It can be seen that combinations 1-4 failed the CharPos test because the CharPos attribute of the character "昨" does not contain the ordered pairs (1, 7), (1, 6), (1, 5), or (1, 4). For combinations 5 and 6, on the other hand, both the CharPos and NextChars tests are passed. The facility therefore looks up combinations 5 and 6 in the lexical knowledge base, to determine that combination 5 is not a word, but combination 6 is a word. After processing combination 6, and determining how far to advance the current position, the facility determines that the IgnoreParts attribute is set, but that the word "昨 天" overlaps with a word candidate beginning with the character "天." The facility therefore advances to the character "天" at the end of combination 6 in accordance with step 613. In combinations 7-12, only combination 12 passes the CharPos and NextChars tests. Combination 12 is therefore looked up and determined to be a word. After processing combination 12, and determining how far to advance the current position, the facility determines that the IgnoreParts attribute of the word constituted by combination 12 is clear, and therefore advances the current position one character to the character "下" rather than to the character following combination 12.

It can further be seen that combinations 18, 24, 37, and 43 are words that have their IgnoreParts attribute set and do not overlap in their final characters with any

word candidates that may be words. After processing each, therefore, the facility advances the current position to the character following the character combination in accordance with step 612, thereby omitting to process unnecessarily up to 41 additional combinations for each of these four combinations.

It can further be seen that the IgnoreParts attributes of the words constituted by combinations 23 and 50 are clear. For this reason, the facility advances the current position only one character in accordance with step 610 after processing these combinations.

It can further be seen that the two-character combinations 30, 36, 47, and 52 are not determined by the facility to constitute words. The facility therefore advances the current position only one character after processing these combinations in accordance with step 610. In all, the facility looks up only 14 of 112 possible combinations in the sample sentence. Of the 14 combinations looked up by the facility, nine are in fact real words.

As shown below in Table 8, after the processing described in conjunction with Table 7, the word list contains the words constituted by combinations 6, 12, 18, 23, 24, 37, 43, 50, and 53.

Number	Word	part of speech
6	昨天	noun
12	天下	noun
18	下午	noun
24	委员	noun
23	委员会	noun
37	布宜诺斯艾利斯	noun
43	讨论	verb
50	这个	pronoun

Number	Word	part of speech
53	问题	noun

Table 8: Word List with Multiple-Character Words

Figure 9 is a flow diagram showing the steps preferably performed by the facility in order to add single-character words to the word list. In steps 901-906, the facility loops through each character in the sentence, from the first character to the last character. In step 902, the facility determines, based on its entry in the lexical knowledge base, whether the character comprises a single-character word, else the facility continues in step 906 without adding a character to the word list. If the character comprises a single-character word, then the facility continues in step 903, else the facility continues in step 906 without adding the character to the word list. In step 903, if the character is contained in a word that may not contain other words, *i.e.*, a word already on the word list has its IgnoreParts attribute set, then the facility continues in step 904, else the facility continues in step 905 to add the character to the word list. In step 904, if the character is contained in a word on the word list that overlaps with another word on the word list, then the facility continues in step 906 without adding the character to the word list, else the facility continues in step 905. In step 905, the facility adds the single-character word comprising the current character to the word list. In step 906, if additional characters remain in the sentence to be processed, then the facility continues in step 901 to process the next character in the sentence, else these steps conclude.

Table 9 below shows that, in performing the steps shown in Figure 9, the facility adds single-character words 54-61 to the word list.

Number	Word	part of speech
6	昨天	noun
54	昨	morpheme
55	天	noun

Number	Word	part of speech
12	天下	noun
56	下	noun (localizer)
18	下午	noun
24	委员	noun
23	委员会	noun
57	会	noun
57	会	verb
58	在	verb
58	在	preposition
58	在	adverb
37	布宜诺斯艾利斯	noun
43	讨论	verb
59	了	function word
50	这个	pronoun
60	这	pronoun
61	个	noun (classifier)
53	问题	noun

Table 9: Word List with Single- and Multiple-Character Words

It should be understood that adding multiple-character words to the word list, and then adding single-character words to the word list is but one exemplary method of creating the word list. In an alternative approach, the word list can be obtained by first locating the single-character words and then adding to the word list multiple-character words. With respect to locating first the single-character words, the approach is similar to the approach described above and illustrated in Figure 9; however, steps 903 and 904 are omitted. Specifically, in step 902, the facility determines, based on its entry in the lexical knowledge base, whether the character comprises a single-character word. If the character

comprises a single-character word, then the facility continues in step 905 to add the character to the word list, else the facility continues in step 906 without adding the character to the word list. The facility processes each character in the sentence to determine if the character is a word by looping through steps 901, 902, 905 and 906.

In the alternative approach, the facility then processes the sentence to locate multiple-character words, and to add such words to the word list. The facility can use the method described above with respect to Figure 6. However, since the sentence may contain multiple-character words that cannot contain other words, i.e., if the IgnoreParts attribute for the multiple-character word is set, then it is beneficial to delete or remove from the word list those single-character words that make up the multiple-character word. Removal of these single-character words from the word list minimizes the analysis required of the parser 133.

The removal of single-character words from the word list is complicated, however, if two multiple-character words, having their IgnoreParts attributes set, overlap. A generic example will be instructive. Suppose, a character sequence ABC is present in the sentence under consideration and that the sequence can comprise multiple character words AB and BC that have their IgnoreParts attribute set. Suppose also that A, B and C are single-character words. There will be a problem if all the single-character words covered by words AB and BC are simply removed from the word list. Specifically, the word A will be missed if BC is the correct word in the sentence. Likewise, the word C will be missed if the word AB is the correct word in the sentence. In either case, the sentence will not be parsed, because none of the "paths" through the sentence is unbroken. To prevent this from happening, all the single-character words in a multiple-character word will be retained regardless of the value of the IgnoreParts attribute except for the word(s) covered by the overlapping part. In the generic example described above, both words A and C will be retained in the word list; however, B will be removed from the word list since it is the overlapping portion of the sequence. Referring to Figure 8, if the facility reaches step 802 in the alternative approach, all of the single-character words

BNSDOCID: <WO___9962001A1_I_>

making up the word candidate would be removed from the list. If the facility, instead, reaches step 806, the non-overlapping single-character words will be retained, while the overlapping portion(s) will be removed.

In the method described above, possible overlapping words are located by examining the NextChar list for the last character in a word candidate (step 801), and ascertaining if a word candidate without its last character is a word (step 804). In an alternative approach, overlapping words can be found by examining other information that is provided to the parser 133 along with the word list. Specifically, in addition to the word list, the parser 133 receives positional information of each word in the word list. From the example of Table 3, each of the characters are numbered sequentially from 1 to 22. Using this positional information, a starting position of the word and an ending position of the word are determined for each word in the word list. Referring to the word identified in Table 9 by way of example, the word denoted by number "6" would have a starting character position of 1 and an ending character position of 2, while the word denoted by number "12" would have a starting character position of 2 and an ending character position of 3. Single-character words would have a starting character position equal to an ending character position. Overlapping words can then be easily ascertained by examining the ending character position and the starting character position of possible adjacent words in the sentence. Specifically, if the ending character position of a possible word in the sentence is greater than or equal to the starting character position of the next possible word in the sentence, an overlap condition exists.

After adding multiple- and single-character words to the word list and generating lexical records for those words, the facility assigns probabilities to the lexical records that is used by the parser to order the application over the lexical records in the parsing process. Figures 10 and 11, discussed below, show two alternative approaches used by the facility in order to assign probabilities to the lexical records.

Figure 10 is a flow diagram showing the steps preferably performed by the facility in order to assign probabilities to the lexical records generated from the words in

the word list in accordance with a first approach. The facility preferably ultimately sets the probability for each lexical record to either a high probability value that will cause the parser to consider the lexical record early during the parsing process, or to a low probability value, which will cause the parser to consider the lexical record later in the parsing process. In steps 1001-1005, the facility loops through each word in the word list. In step 1002, if the current word is contained in a larger word in the word list, then the facility continues in step 1004, else the facility continues in step 1003. In step 1003, the facility sets the probability for the lexical record representing the word to the high probability value. After step 1003, the facility continues in step 1005. In step 1004, the facility sets the probability for the lexical records representing the word to the low probability value. After step 1004, the facility continues in step 1005. In step 1005, if additional words remain in the word list to be processed, then the facility continues in step 1001 to process the next word in the word list, else these steps conclude.

Table 10 below shows the probability values assigned to each word in the word list in accordance with steps shown in Figure 10. It can be seen by reviewing the probabilities that the facility assigns the high probability value to at least one word containing each character, so that at least one lexical record containing each character is considered early in the parsing process.

Number	Word	part of speech	probability value
6	昨天	noun	high
54	昨	morpheme	low
55	天	noun	low
12	天下	noun	high
56	下	noun (localizer)	low
18	下午	noun	high
24	委员	noun	low
23	委员会	noun	high

Number	Word	part of speech	probability value
57	会	noun	low
57	会	verb	low
58	在	verb	high
58	在	preposition	high
58	在	adverb	high
37	布宜诺斯艾利斯	noun	high
43	讨论	verb	high
59	了	function word	high
50	这个	pronoun	high
60	这	pronoun	low
61	个	noun (classifier)	low
53	问题	noun	high

Table 10: Word List with Probabilities

Figure 11 is flow diagram showing the steps preferably performed by the facility in order to assign probabilities to the lexical records generated from the words in the word list in accordance with a second approach. In step 1101, the facility uses the word list to identify all the possible segmentations of the sentence comprised entirely of the words in the word list. In step 1102, the facility selects the one or more possible segmentations identified in step 1101 that contain the fewest words. If more than one of the possible segmentations has the minimum number of words, the facility selects each such possible segmentation. Table 11 below shows the possible segmentation generated from the word list shown in Table 9 having the fewest words (9).

<u>昨天</u> <u>下午</u> <u>委员会</u> <u>在</u> <u>布宜诺斯艾利斯</u> <u>讨论</u> <u>了</u> <u>这个</u> <u>问题</u>。

Table 11

In step 1103, the facility sets the probability for the lexical records of the words in the selected segmentation(s) to the high probability value. In step 1104, the facility sets the probability for the lexical records of the words not in selected segmentation(s) to the low probability value. After step 1104, these steps conclude.

Table 12 below shows the probability values assigned to each word in the word list in accordance with steps shown in Figure 11. It can be seen by reviewing the probabilities that the facility assigns the high probability value to at least one word containing each character, so that at least one lexical record containing each character is considered early in the parsing process.

Number	Word	part of speech	probability value
6	昨天	Noun	high
54	昨	Morpheme	low
55	天	Noun	low
12	天下	Noun	low
56	下	noun (localizer)	low
18	下午	Noun	high
24	委员	noun	low
23	委员会	noun	high
57	会	noun	low
57	会	Verb	low
58	在	Verb	high
58	在	Preposition	high
58	在	Adverb	high
37	布宜诺斯艾利斯	Noun	high
43	讨论	Verb	high
59	了	Function word	high
50	这个	Pronoun	high

Number	Word	part of speech	probability value
60	这	Pronoun	low
61	个	Noun (classifier)	low
53	问题	Noun	high

Table 12: Word List with Probabilities

In one broad aspect of the present invention, probabilities can also be assigned to overlapping pairs of words. In the generic character sequence ABC statistical data may indicate that the probability of the combination of words AB and C is higher than the combination of A and BC. Thus, the parser 133 should consider the combination AB and C first, whereas the combination of A and BC should not be considered unless no successful analysis can be found using AB and C. Statistical data may also indicate that one of the possible combinations AB and C, or A and BC is impossible.

In order to assign relative probabilities to a word in an overlapping pair of words, or remove impossible combinations, information is stored in the lexical knowledge base 132. In particular, additional lists can be associated with many multiple-character words in the lexical knowledge base 132. The lists include:

(1) a first left condition list – the word in this entry would be assigned a low probability if it is immediately preceded by one of the characters in this list in the sentence;

(2) a first right condition list – the word in this entry would be assigned a low probability if it is immediately followed by one of the characters in this list in a sentence;

(3) a second left condition list – the word in this entry would be ignored if it is immediately preceded by one of the characters in this list in a sentence. In other words, if a multiple-character word in the word list meets this condition, it will be removed from the word list; and

(4) a second right condition list – the word in this entry would be ignored if it is immediately followed by one of the characters in this list in a sentence. In other words, if the word in the word list meets this condition, it will be removed from the word list.

It should be noted that each of the foregoing lists may not be present for every multiple-character word in the lexical knowledge base 132. In other words, some of the multiple character words in the lexical knowledge base 132 may not have any of the foregoing lists, while other will have one, some or all of the lists. If desired, other lists can be generated based on immediately preceding or following characters. For instance, lists can be generated to assign high probabilities. The lists are entered in the lexical knowledge base 132 manually.

In addition to analysis using a lexical knowledge base to resolve disambiguation as discussed above, a rule-base disambiguation analysis can also be used in combination with the lexical analysis before parsing begins. For example, if a character string ABCD is present in a sentence where AB, BC and CD are all possible words, word BC can be ignored (removed from the word list) if AB does not overlap with a preceding word, CD does not overlap with a following word, either A or D is a non-word, and neither ABC nor BCD is a word.

It should be emphasized, however, that there is no logical dependency between the parser's ability to resolve segmentation ambiguities and the lexical disambiguation described above. The elimination of words at the lexical level reduces parsing complexity, but is not always a necessary condition for the successful analysis of a sentence. Parsing will be successful as long as all of the correct words in a sentence are

in the word list provided by the facility 131, and the number of words in the word list is not so great as to overburden the parser 133. Therefore, the success of sentence analysis, including correct word segmentation, does not depend on the complete success of lexical disambiguation, though the latter will greatly facilitate the former. This allows development of the facility 131 and the parser 133 independently despite the fact that there is interaction between the components.

Figure 12 is a parse tree diagram showing a parse tree generated by the parser representing the syntactic structure of the sample sentence. It can be seen that the parse tree is a hierarchical structure having a single sentence record 1231 as its head and having a number of lexical records 1201-1211 as its leaves. The parse tree further has intermediate syntactic records 1221-1227 that combine lexical records each representing a word into a larger syntactic structure representing one or more words. For example, the prepositional phrase record 1223 combines a lexical record 1204 representing a preposition and lexical record 1206, representing a noun. In accordance with step 506 of Figure 5, the facility identifies the words represented by lexical records 1201-1211 in the parse tree as the words into which the sample sentence should be segmented. This parse tree may also be retained by the facility in order to perform additional natural language processing on the sentence.

While this invention has been shown and described with reference to preferred embodiments, it will be understood by those skilled in the art that various changes or modifications in form and detail may be made without departing from the scope of the invention. For example, aspects of the facility may be applied to perform word segmentation in languages other than Chinese. Further, proper subsets or supersets of the techniques described herein may be applied to perform word segmentation.

CLAIMS

We claim:

1. A method in a computer system for identifying individual words occurring in a sentence of text, the method comprising the steps of:

for each of a plurality of words:

storing an indication of probability of whether the word occurs in natural language text as a function of adjacent characters;

for each of a plurality of contiguous groups of characters occurring in the sentence:

determining overlapping possible words;

ascertaining probability based on the stored indication and adjacent characters; and

submitting the groups of characters determined to be possible words to a parser with an indication of probability.

2. The method of claim 1 wherein for each of a plurality of words having an indication of probability, the data structure further comprises an associated list of characters.

3. The method of claim 1 wherein an indication of probability is low if the word is preceded by one of the characters in a list.

4. The method of claim 1 wherein an indication of probability is low if the word is followed by one of the characters in a list.

5. The method of claim 1 wherein an indication of probability is zero if the word is preceded by one of the characters in a list.

6. The method of claim 1 wherein an indication of probability is zero if the word is followed by one of the characters in a list.

7. The method of claim 1 wherein the natural language is Chinese.

8. A computer-readable medium storing instructions for a computer system for identifying individual words occurring in a sentence of text, the instructions comprising the steps of:

for each of a plurality of words:

storing an indication of probability of whether the word occurs in natural language text as a function of adjacent characters;

for each of a plurality of contiguous groups of characters occurring in the sentence:

determining overlapping possible words;

ascertaining probability based on the stored indication and adjacent characters; and

submitting the groups of characters determined to be possible words to a parser with an indication of probability.

9. The computer-readable medium of claim 8 wherein for each of a plurality of words having an indication of probability, the data structure further comprises an associated list of characters.

10. The computer-readable medium of claim 8 wherein an indication of probability is low if the word is preceded by one of the characters in a list.

11. The computer-readable medium of claim 8 wherein an indication of probability is low if the word is followed by one of the characters in a list.

12. The computer-readable medium of claim 8 wherein an indication of probability is zero if the word is preceded by one of the characters in a list.

13. The computer-readable medium of claim 8 wherein an indication of probability is zero if the word is followed by one of the characters in a list.

14. The computer-readable medium of claim 8 wherein the natural language is Chinese.

15. A computer memory containing a word segmentation data structure for use in identifying individual words occurring in natural language text, the data structure comprising:

for each of a plurality of words:

an indication of probability of whether the word occurs in natural language text as a function of adjacent characters.

16. The computer memory of claim 15 wherein for each of a plurality of words having an indication of probability, the data structure further comprises an associated list of characters.

17. The computer memory of claim 15 wherein an indication of probability is low if the word is preceded by one of the characters in a list.

18. The computer memory of claim 15 wherein an indication of probability is low if the word is followed by one of the characters in a list.

19. The computer memory of claim 15 wherein an indication of probability is zero if the word is preceded by one of the characters in a list.

20. The computer memory of claim 15 wherein an indication of probability is zero if the word is followed by one of the characters in a list.

21. A computer memory containing a word segmentation data structure for use in identifying individual words occurring in natural language text, the data structure comprising:

for each of a plurality of characters:

an identification of characters that occur in the second position of words that begin with the character, and

for words containing the character:

an identification of the length of the word and the character position within the word occupied by the character; and

for each of a plurality of words:

an indication of whether the sequence of characters that comprises the word may also comprise a series of shorter words; and

an indication of probability of whether the word occurs in natural language text as a function of adjacent characters.

22. The computer memory of claim 21 wherein for each of a plurality of words having an indication of probability, the data structure further comprises an associated list of characters

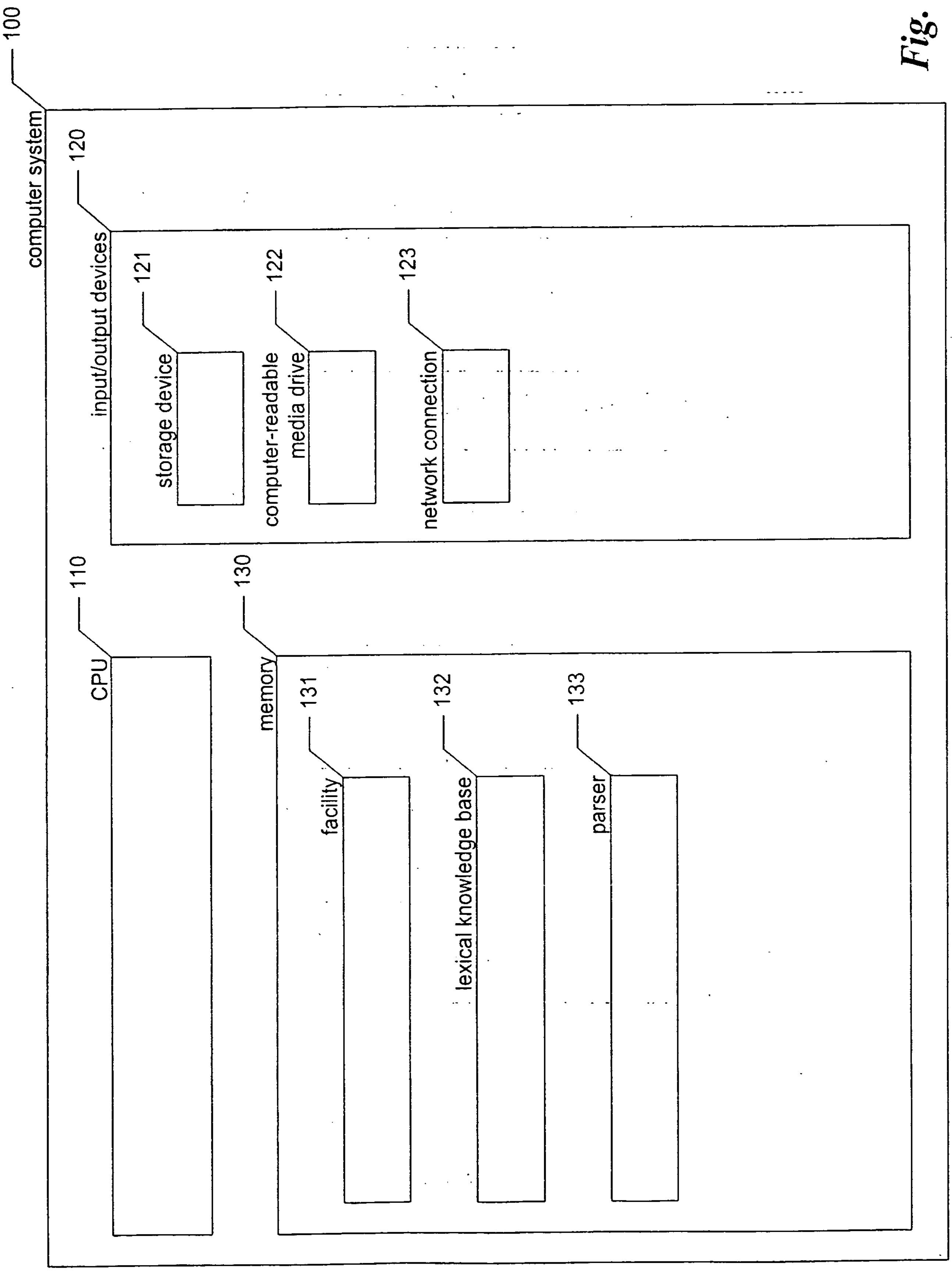

100
computer system
120
input/output devices
121
storage device
122
computer-readable media drive
123
network connection
110
CPU
130
memory
131
facility
132
lexical knowledge base
133
parser

Fig. 1

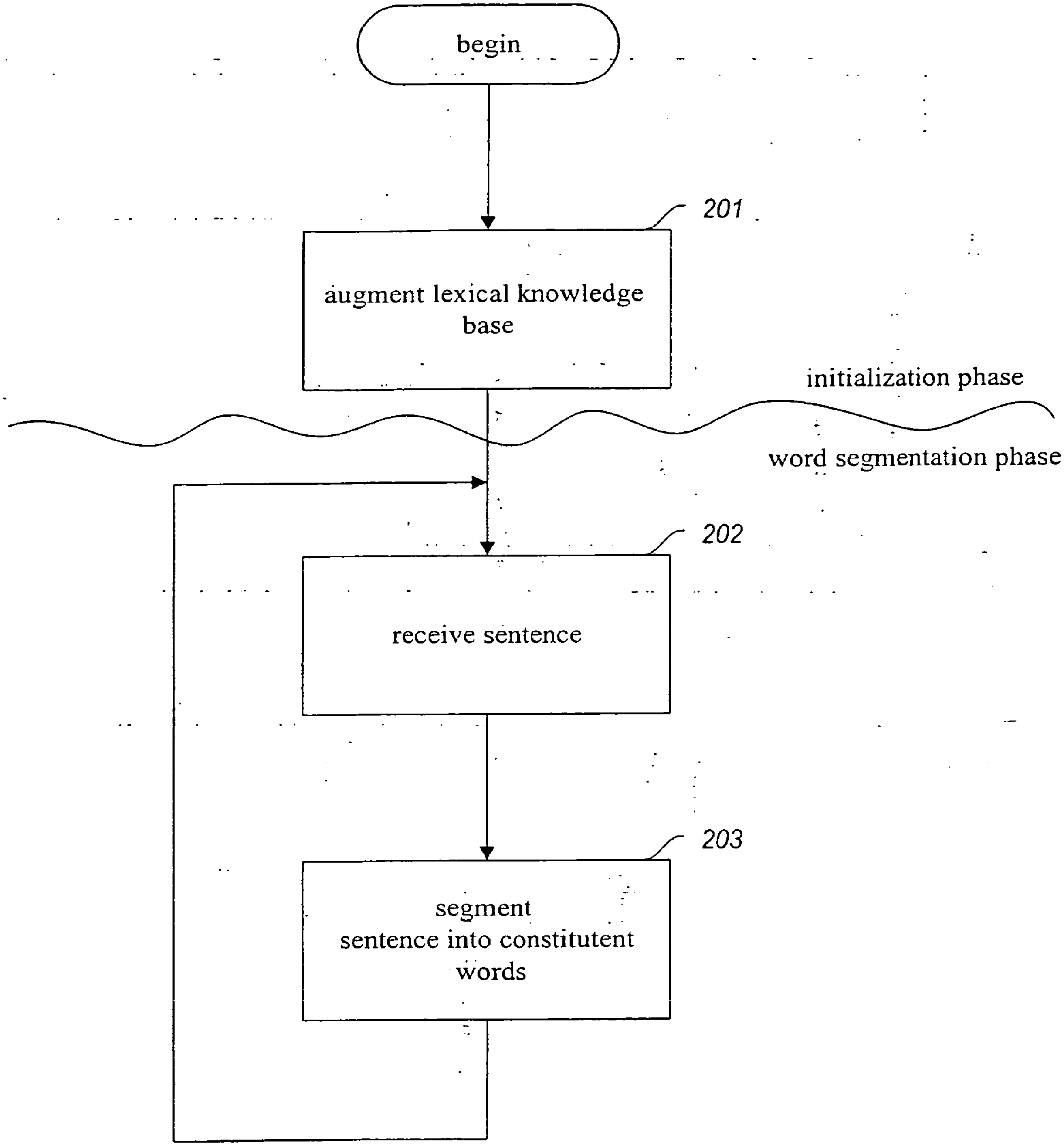


Fig. 2

BNSDOCID: <WO___9962001A1_I_>

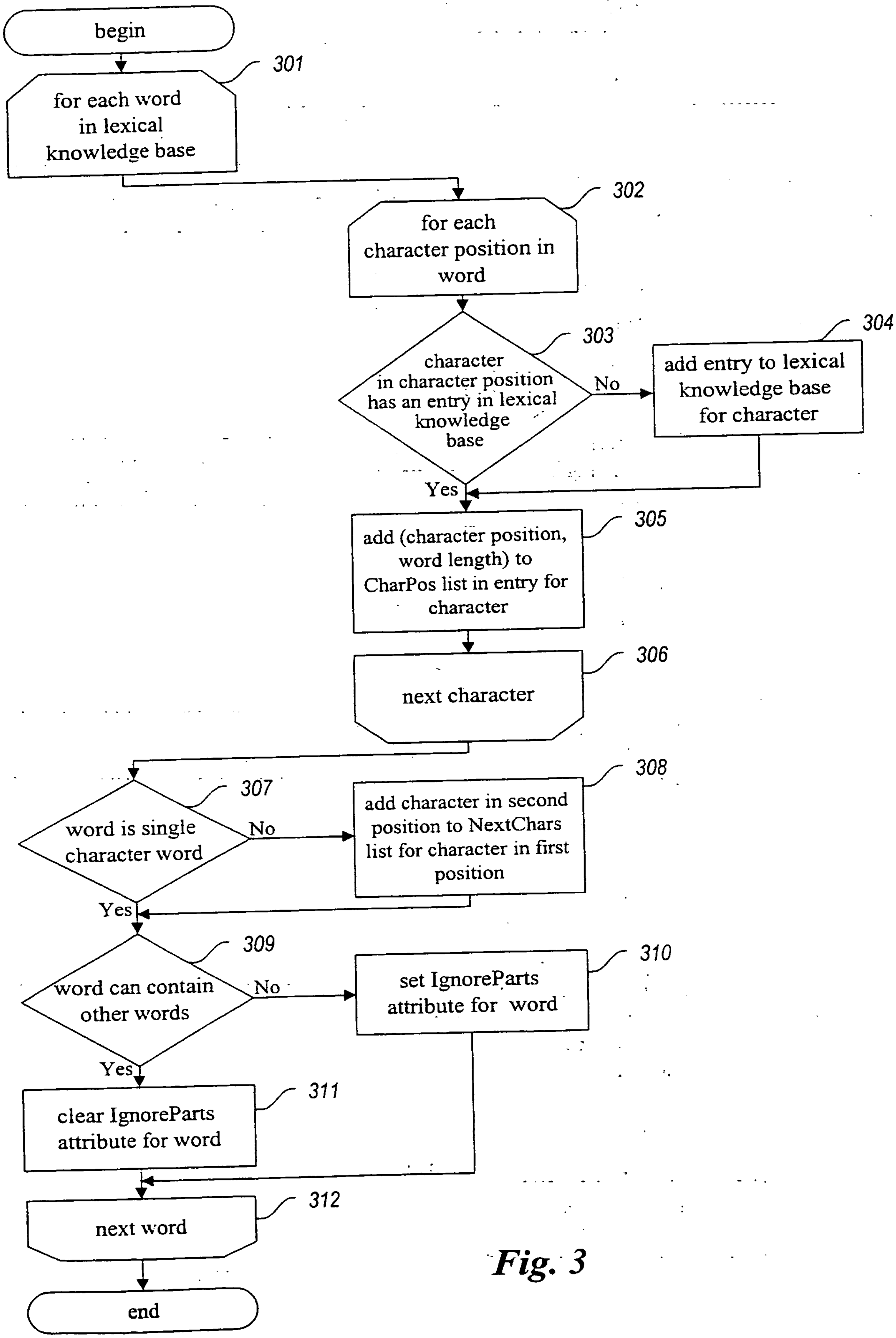

begin
for each word in lexical knowledge base
301
for each character position in word
302
character in character position has an entry in lexical knowledge base
303
No
add entry to lexical knowledge base for character
304
Yes
add (character position, word length) to CharPos list in entry for character
305
next character
306
word is single character word
307
No
add character in second position to NextChars list for character in first position
308
Yes
word can contain other words
309
No
set IgnoreParts attribute for word
310
Yes
clear IgnoreParts attribute for word
311
next word
312
end

Fig. 3

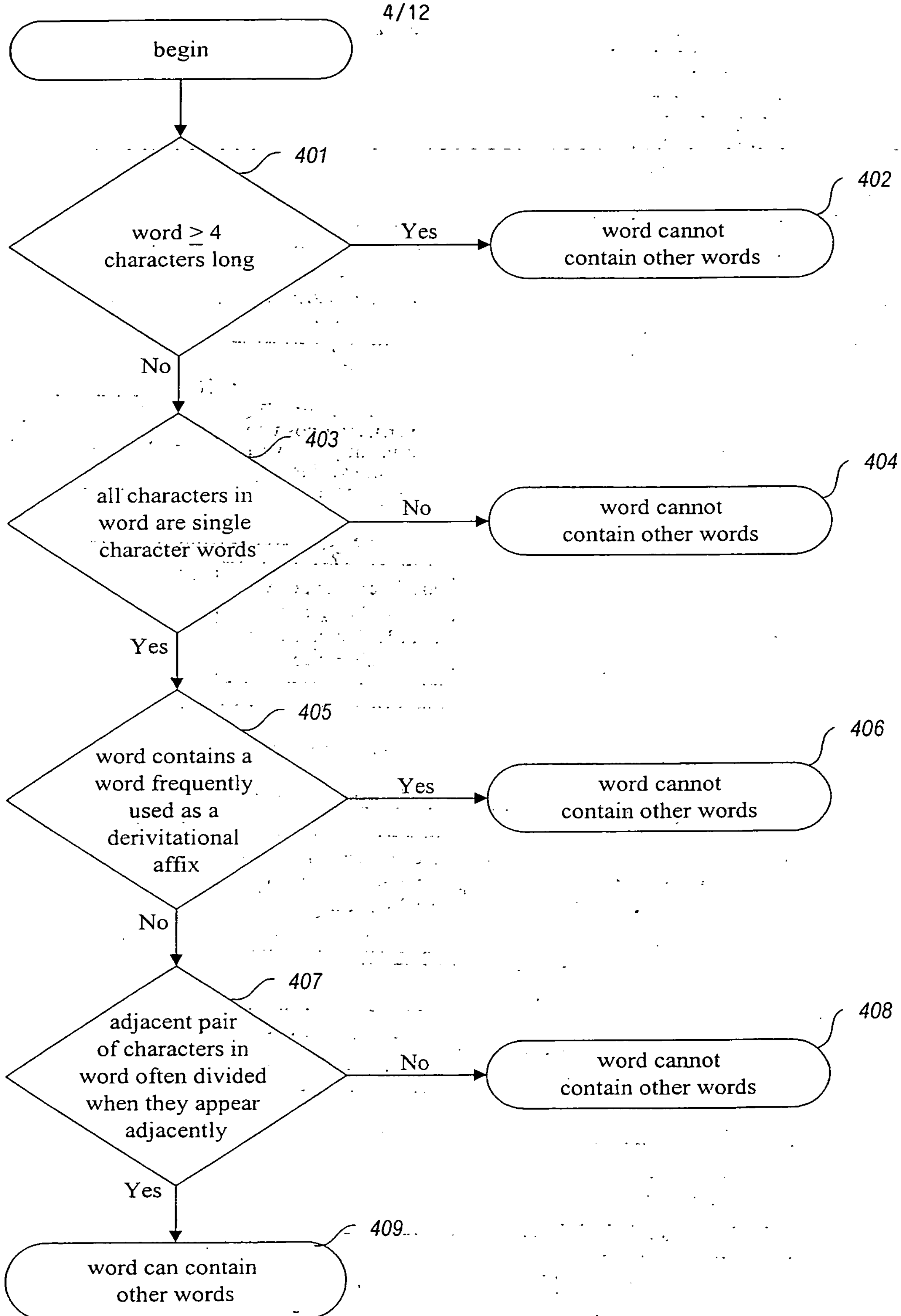

begin
401
word ≥ 4 characters long
Yes
402
word cannot contain other words
No
403
all characters in word are single character words
No
404
word cannot contain other words
Yes
405
word contains a word frequently used as a derivitational affix
Yes
406
word cannot contain other words
No
407
adjacent pair of characters in word often divided when they appear adjacently
No
408
word cannot contain other words
Yes
409
word can contain other words

Fig. 4

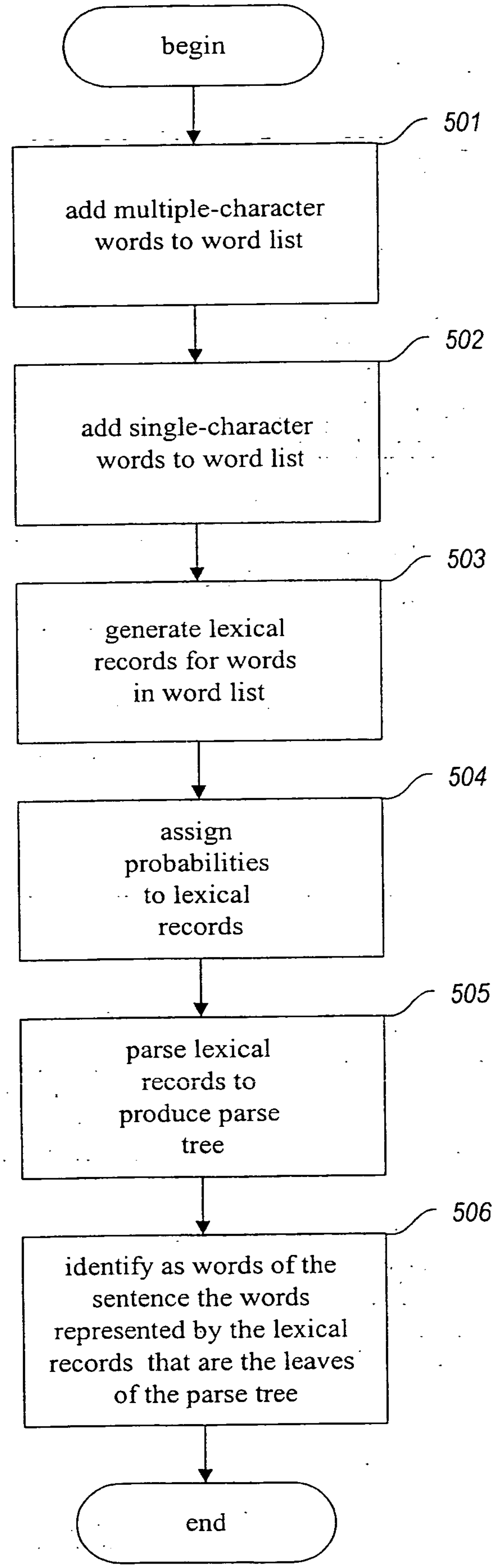

begin
501
add multiple-character words to word list
502
add single-character words to word list
503
generate lexical records for words in word list
504
assign probabilities to lexical records
505
parse lexical records to produce parse tree
506
identify as words of the sentence the words represented by the lexical records that are the leaves of the parse tree
end

Fig. 5

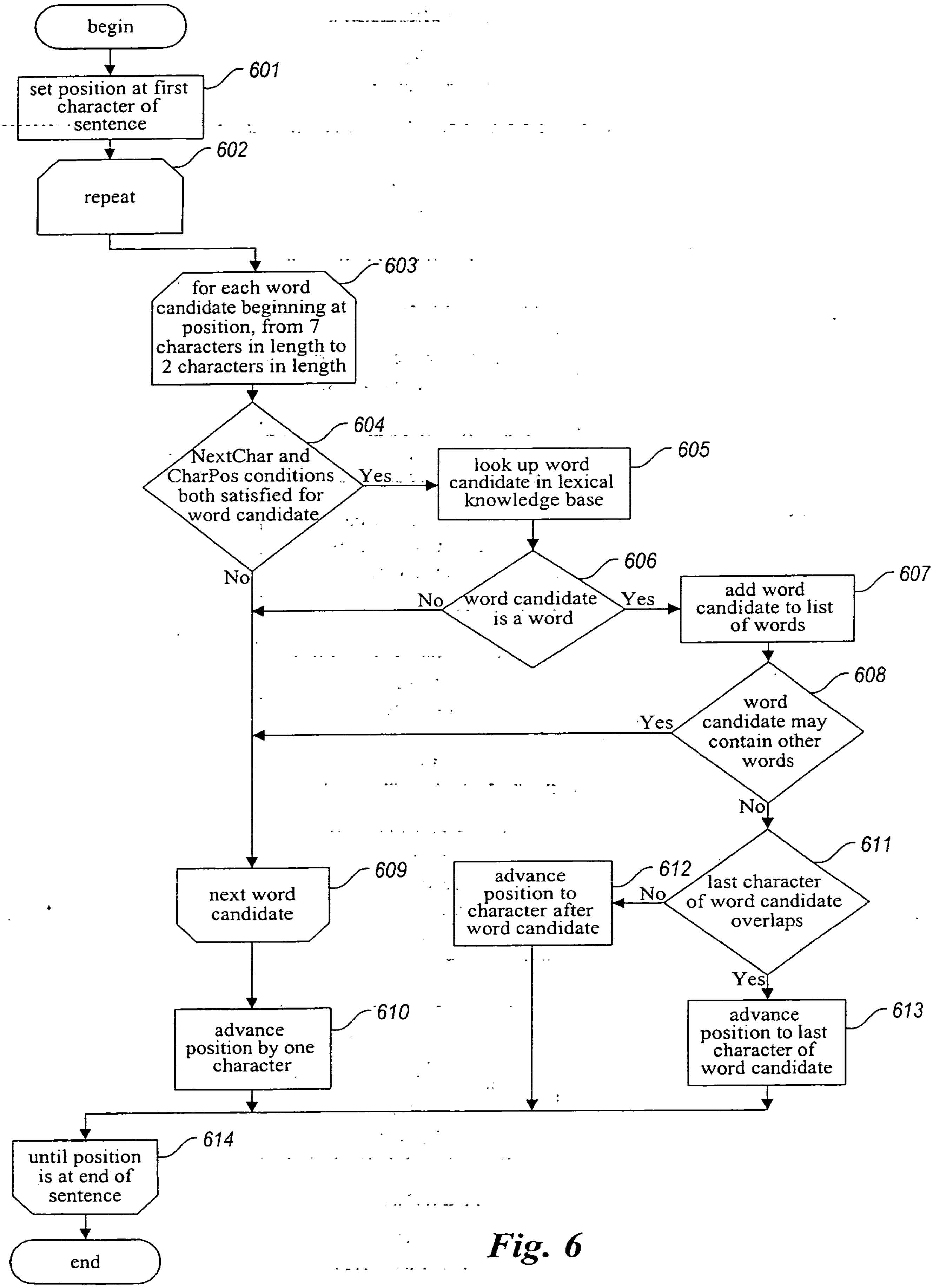

begin
set position at first character of sentence
601
repeat
602
for each word candidate beginning at position, from 7 characters in length to 2 characters in length
603
NextChar and CharPos conditions both satisfied for word candidate
604
Yes
No
look up word candidate in lexical knowledge base
605
word candidate is a word
606
No
Yes
add word candidate to list of words
607
word candidate may contain other words
608
Yes
No
last character of word candidate overlaps
611
No
Yes
next word candidate
609
advance position to character after word candidate
612
advance position by one character
610
advance position to last character of word candidate
613
until position is at end of sentence
614
end

Fig. 6

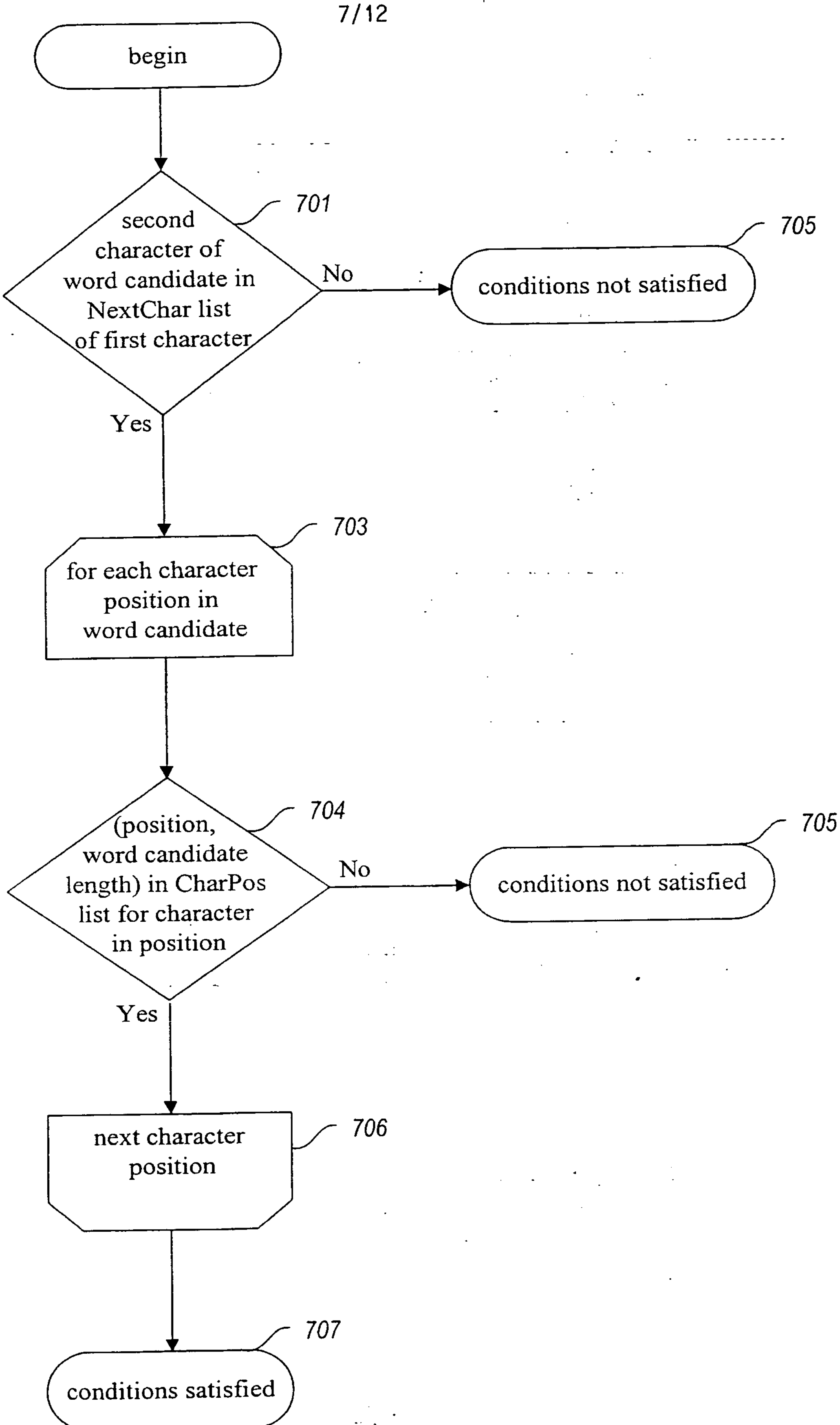

begin
second character of word candidate in NextChar list of first character
701
No
conditions not satisfied
705
Yes
for each character position in word candidate
703
(position, word candidate length) in CharPos list for character in position
704
No
conditions not satisfied
705
Yes
next character position
706
conditions satisfied
707

Fig. 7

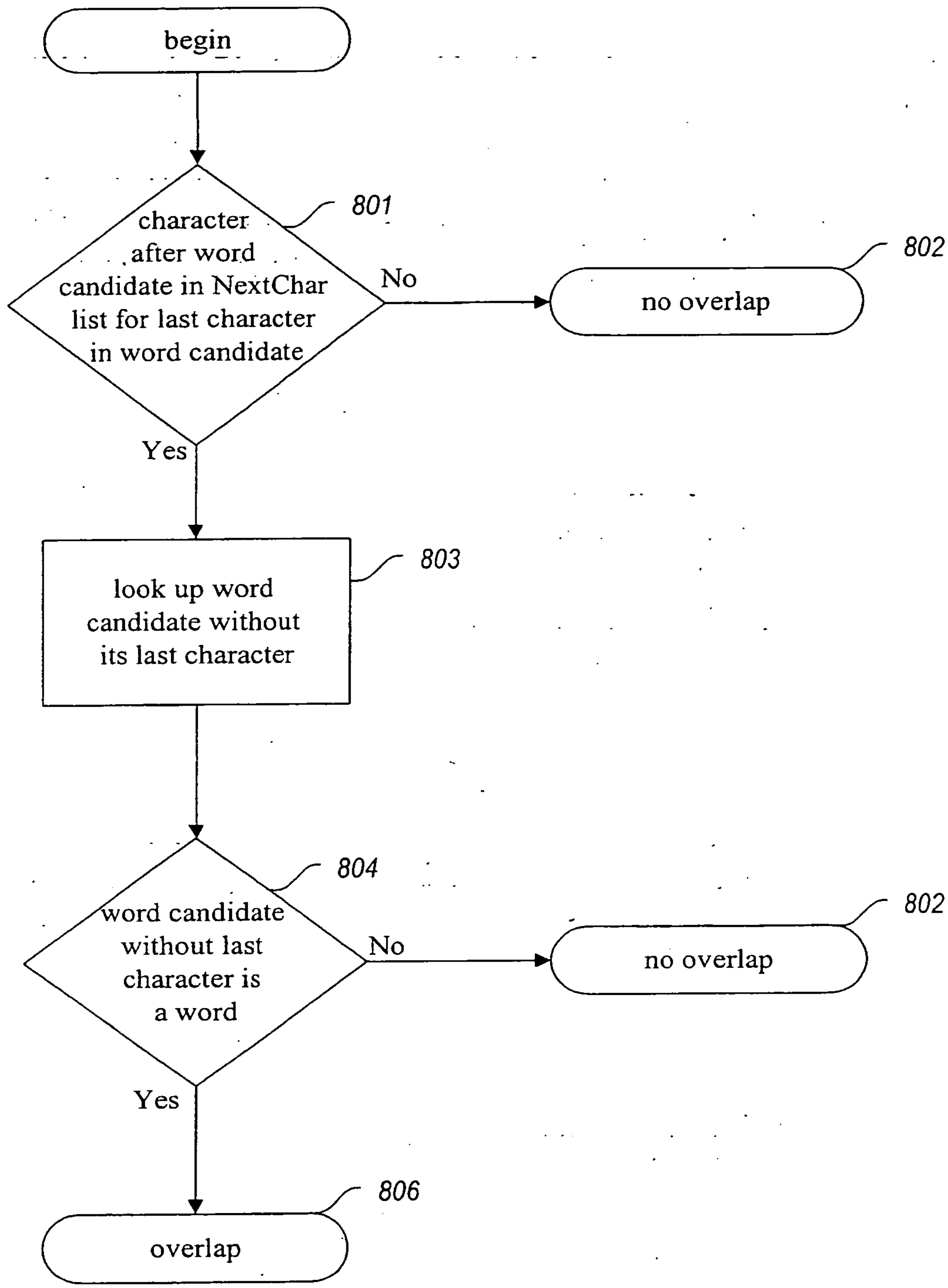

begin
character after word candidate in NextChar list for last character in word candidate
801
No
no overlap
802
Yes
look up word candidate without its last character
803
word candidate without last character is a word
804
No
no overlap
802
Yes
overlap
806

Fig. 8

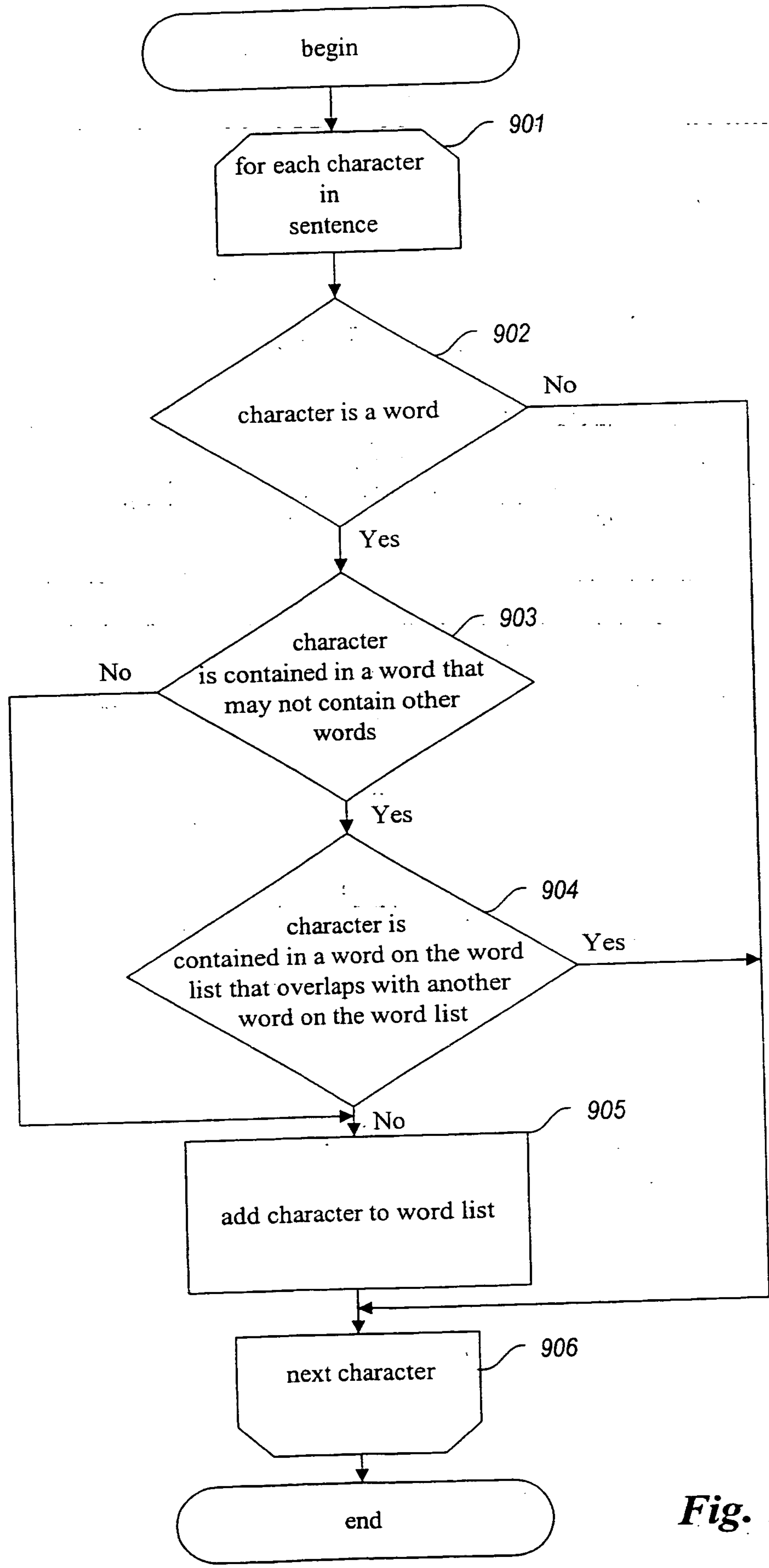

begin
for each character in sentence
901
character is a word
902
No
Yes
character is contained in a word that may not contain other words
903
No
Yes
character is contained in a word on the word list that overlaps with another word on the word list
904
Yes
No
add character to word list
905
next character
906
end

Fig. 9

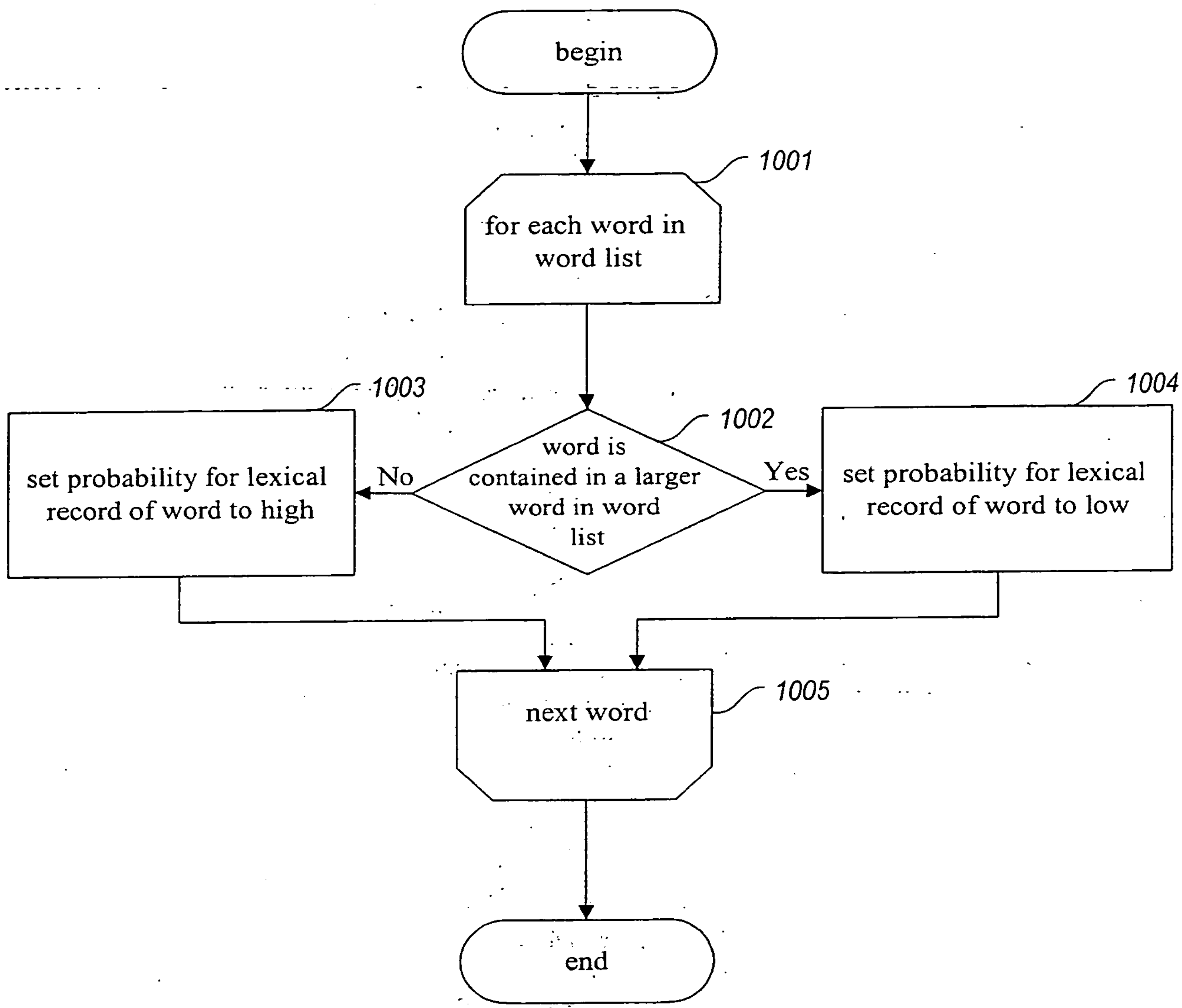

begin
1001
for each word in word list
1003
1002
1004
set probability for lexical record of word to high
No
word is contained in a larger word in word list
Yes
set probability for lexical record of word to low
1005
next word
end

Fig. 10

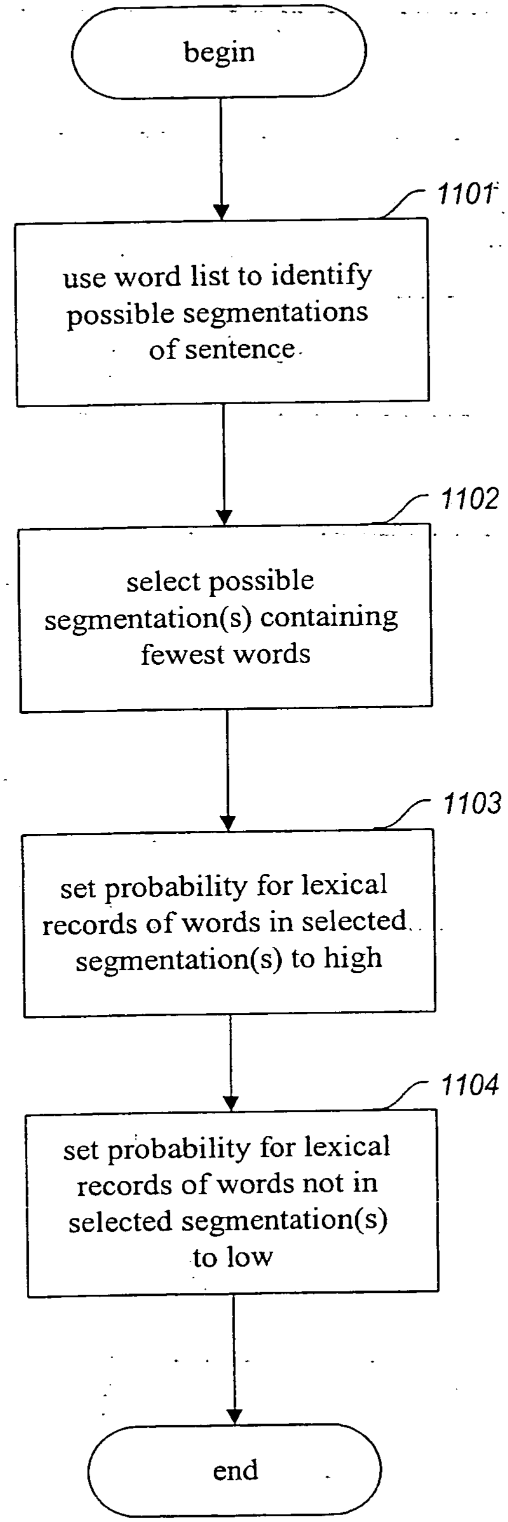

begin
1101
use word list to identify possible segmentations of sentence
1102
select possible segmentation(s) containing fewest words
1103
set probability for lexical records of words in selected segmentation(s) to high
1104
set probability for lexical records of words not in selected segmentation(s) to low
end

Fig. 11

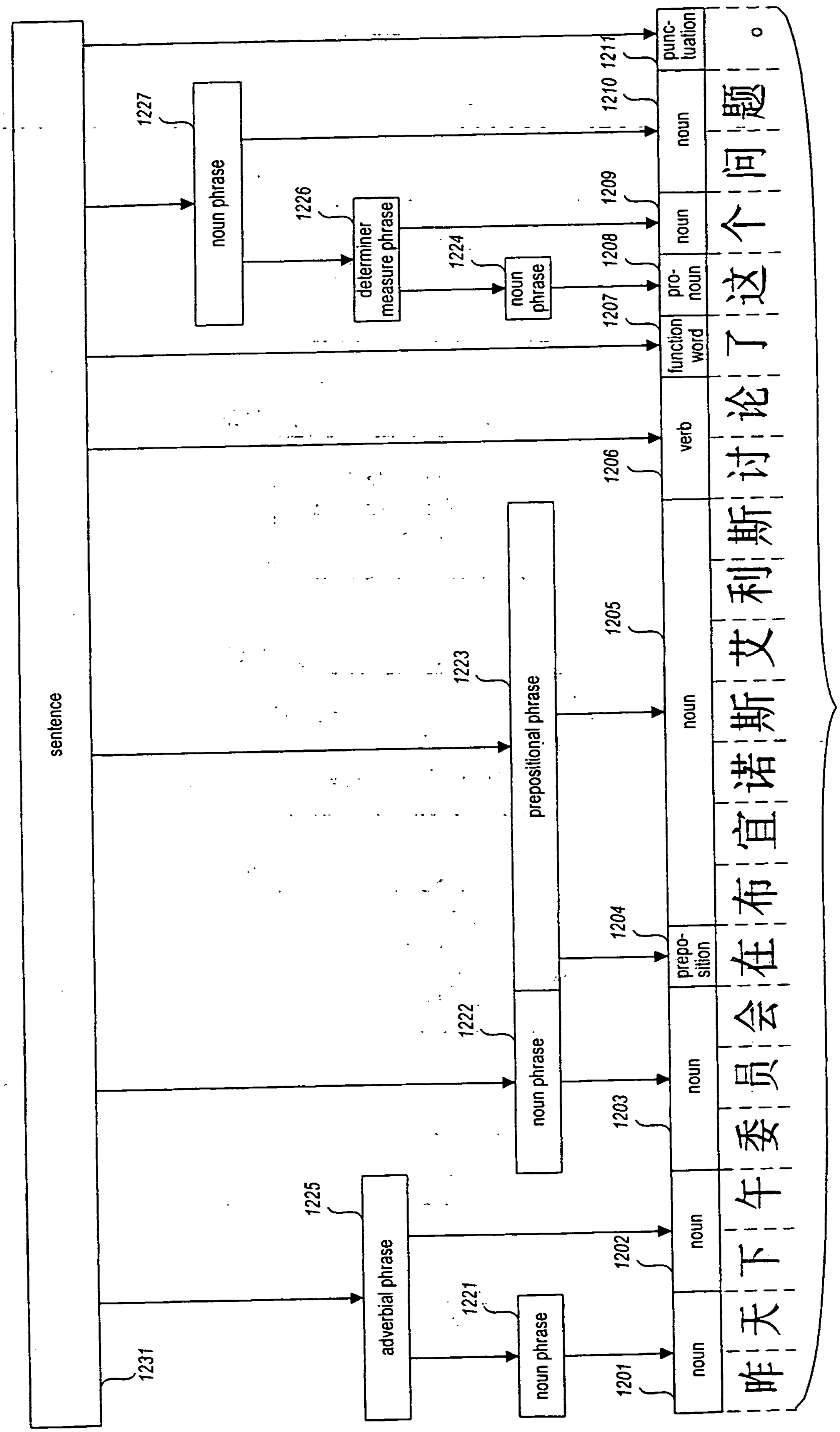

sentence
1231
adverbial phrase
1225
noun phrase
1221
noun phrase
1222
prepositional phrase
1223
noun phrase
1224
determiner measure phrase
1226
noun phrase
1227
noun
noun
noun
prepo-sition
noun
verb
function word
pro-noun
noun
noun
punc-tuation
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
昨天下午委员会在布宜诺斯艾利斯讨论了这个问题。
1200

Fig. 12

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 99/11856

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F17/27

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CHING-LONG YEH ET AL: "Rule-based word identification for Mandarin Chinese sentences a unification approach" COMPUTER PROCESSING OF CHINESE & ORIENTAL LANGUAGES, MARCH 1991, USA, vol. 5, no. 2, pages 97-118, XP002116761 ISSN: 0715-9048 page 100, line 17 -page 113, line 6 ---	1-22

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
28 September 1999	13/10/1999

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Pedersen, N

1

Form PCT/ISA/210 (second sheet) (July 1992)

BNSDOCID: <WO___9962001A1_I_>

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 99/11856

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JIAN-YUN NIE ET AL: "Unknown word detection and segmentation of Chinese using statistical and heuristic knowledge" COMMUNICATIONS OF COLIPS, DEC. 1995, COLIPS, CHINESE & ORIENTAL LANGUAGES INF. PROCESS. SOC., SINGAPORE, vol. 5, no. 1-2, pages 47-57, XP002116762 ISSN: 0218-7019 page 49, column 1, line 1 -page 53, column 2, line 8	1-22
A	WO 98 08169 A (INSO CORP) 26 February 1998 (1998-02-26) abstract	1-22
A	TELLER V ET AL: "A PROBABILISTIC ALGORITHM FOR SEGMENTING NON-KANJI JAPANESE STRINGS" PROCEEDINGS NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, vol. 1, 31 July 1994 (1994-07-31), pages 742-747, XP000612334 page 743, column 1, line 6 - line 32 page 744, column 2, line 2 -page 745, column 1, line 53	1-22
A	ANONYMOUS: "Method of Segmenting Texts into Words" IBM TECHNICAL DISCLOSURE BULLETIN, vol. 39, no. 11, pages 115-118, XP000679841 New York, US page 1, line 1 - line 42	1-22
A	US 5 448 474 A (ZAMORA ANTONIO) 5 September 1995 (1995-09-05) column 2, line 50 -column 4, line 11	1-22

1

BNSDOCID: <WO___9962001A1_I_>

INTERNATIONAL SEARCH REPORT

.formation on patent family members

Interr ·onal Application No
PCT/US 99/11856

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9808169 A	26-02-1998	NONE	
US 5448474 A	05-09-1995	CN 1100542 A	22-03-1995
		JP 2741835 B	22-04-1998
		JP 6325076 A	25-11-1994
		KR 122518 B	20-11-1997

Form PCT/ISA/210 (patent family annex) (July 1992)

BNSDOCID: <WO___9962001A1_I_>